Method and Apparatus for changing the output delay of audio or video data encoding

FIELD OF THE INVENTION

5 The invention relates to a method for changing the output delay of audio or video data encoding and to an apparatus for changing the output delay of audio or video data encoding.

10

BACKGROUND OF THE INVENTION

Encoding systems with related video encoders and audio encoders are used for various applications, e.g. for TV broadcasting purposes. In this case the video encoders can have

15 variable encoding delay of up to 1.5 sec, depending for example upon the selected bit rate. Therefore, the audio and video delay should be aligned before multiplexing and transmitting the audio and video streams, because of buffer limitations in the decoders of consumer products.

20

A basic mechanism for controlling delay within the audio encoder by help of a time stamping mechanism can be found in the European patent application 99250009. In a multi-channel audio encoder board input time stamps are generated which

25 become linked at least in one input processing stage with frames of audio data to be encoded, wherein the input time stamps or time stamps derived from the input time stamps remain linked with the correspondingly processed frame data in the different processing stages in the processing, but are

30 at least in the last processing stage replaced by output time stamps. In each of theses stages the corresponding time stamp information linked with current frame data to be processed is regarded in order to control the overall delay of the processing.

35

In order to allow switchable bit rates for the video and audio encoders on operators choice, for example for making space for an additional TV channel, a switchable delay of

2

the audio encoder is desireable. However, the European pat-
ent application 99250009 does not disclose, how such delay
changes might be handled.

5

SUMMARY OF THE INVENTION

It is one object of the invention to disclose a method for
changing the output delay of audio or video data encoding,
particularly for the purpose of switchable bit rates for the
10   video and audio encoders on operators choice. This object is
achieved by the method disclosed in claim 1.

It is a further object of the invention to disclose an appa-
ratus for changing the output delay of audio or video data
15   encoding which utilises the inventive method. This object is
achieved by the apparatus disclosed in claim 7.

In principle, according to the inventive method input time
stamps are generated which become linked with audio or video
20   data to be encoded and are used to control the delay of the
encoding process. Output time stamps are derived from the
input time stamps by using a data delay constant and are as-
signed to the encoded data for indicating the output time.
The encoded data with assigned output time stamps are buff-
25   ered before output, wherein for a change of the output delay
the data delay constant is changed. Already assigned output
time stamps remain unchanged. For data for which output time
stamps are not already assigned, the output time stamps are
calculated using the new data delay constant.

30

Advantageous additional embodiments of the inventive method
are disclosed in the respective dependent claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments of the invention are described with reference to the accompanying drawings, which show in:

5 Fig. 1 a schematic flow chart of the method for changing the output delay;

Fig. 2 a functional block diagram of a 4-channel audio encoder using the inventive method.

10

DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

Fig. 1 shows a schematic flow chart of the method for changing the output delay. To audio samples or audio coefficients an input time stamp information ITS is linked or assigned in

15 the first method step 1. Then the audio samples or audio coefficients are encoded in method step 2. In method step 3 it is checked, whether a change of the user-defined delay D is requested. If so, the processing equation for the output time stamps $OTS = ITS + D$ is changed in method step 4.

20 The input time stamp information ITS is then replaced by the output time stamps OTS in method step 5 before buffering in method step 6. The buffering is performed at the output of the encoding process, as the amount of memory required is lower at this side. Thus input data will typically be imme-

25 diately encoded when received by the audio encoder, and be put after encoding into the delay buffer, especially in form of transport stream packets.

Before sending the data to the output in method step 9, the

30 OTS are checked in method step 7. If a gap in the OTS continuity occurs, stuffing data or zero data are inserted in method step 8. If on the other hand two packets with same or overlapping OTS are found in the delay buffer, this also requires a special treatment in method step 8. One possibility

35 is to discard packets indicating output times that are already passed. One other possibility to handle this case is to write no further data into the output delay buffer beginning with the change request for the difference time, and to

use the new delay time for the OTS calculation for all fol-
lowing packets. Finally, the data are sent to the output in
method step 9.

5 In the following the treatment of the data, especially in
method step 8, is described in more detail for an example
delay increase or reduction, respectively, of 0.3 sec and an
initial delay of 0.5 sec.

10 For delay increase the delay buffer contains a certain
amount of data equal to the presently effective delay time
of 0.5 sec and this amount of data shall be increased to 0.8
sec. This implies, that the output of encoded data from the
delay buffer effectively needs to be stopped for 0.3 sec
15 while data input to the delay buffer is continued.

The data in the delay buffer are already encoded data repre-
senting the continuation of the delay buffer already output
just before. Therefore, the delay buffer is managed in the
20 way, that after the change request for the delay time, the
delay buffer continues to deliver data for 0.5 sec until all
data that were in the delay buffer at the time of the delay
time change are completely output. Then the output from the
delay buffer is stopped, implying that either stuffing,
25 zero data or no packets are sent to the transmitter/decoder
chain. The stop then will last for a time of 0.3 sec, which
is required to increase the delay buffer contents accord-
ingly.

30 This behavior can be accomplished by use of the above men-
tioned time stamp based delay control mechanism. All output
blocks, i.e. TS packets, that reside in the delay buffer at
a given time are also stamped with an output time stamp that
indicates the point in time when the packet should be sent
35 out of the delay buffer and afterwards to the transmitter.
Nothing needs to be changed for the packets that are already
in the delay buffer, they are output as intended at their
generation time. Immediately at the time where the audio de-

lay D is changed, either directly or indirectly by operator, the processing equation for the output time stamps OTS = ITS + D is changed, i.e. all OTS time stamps are increased by 0.3 sec. All packets computed with the "old" delay are out-

5   put orderly one after another by the output stage from the delay buffer. Then, after 0.5 sec, there will be a gap in the OTS continuity, i.e. the next packet will indicate an OTS that is indicating a time increased by 0.3 sec than the packet would have had without a delay change. The output

10   stage then can send stuffing or zero data or even no pack-ets.

Thus the net effect of delay increase for the user, i.e. the consumer listening at the decoder side, will be, that after

15   the delay change request
     - the audio program does continue normal for the present ef-fective delay time of 0.5 sec,
     - the audio program shortly mutes for 0.3 sec, and
     - the program continues normally with the new delay of 0.8

20   sec.

On the operator side, upon delay change request the follow-ing happens
     - all program parts already input to the encoder will be se-

25   quentially delivered to the user,
     - all audio program parts input after the delay switch will be separated by a short break from the previous parts on the user side.

30   The operator could use a program gap or the moment of a switch between distinct program parts for the delay time change in order to achieve a minimum irritation for the user.

35   Also at delay reduction the delay buffer contains a certain amount of data. The delay shall now be reduced from 0.5 sec to 0.2 sec delay. In this case, it will be necessary to normally continue the outputting process for 0.3 sec, while

stopping the write of further input data into the delay
buffer. Thus a short time of program material available at
the input of the encoder will not be sent to the user. In
principle the audio program at the user side could be con-
5   tinuous but a short piece would be cut out of the signal.

The delay reduction can be accomplished by the same time
stamp based delay control. Immediately after the audio delay
change request, the output time stamp OTS calculation is
10 changed in a way, that the OTS indicate a point in time ear-
lier by 0.3 sec then without the delay change.

If the writing of data packets into the delay buffer is con-
tinued, this would result in finding two packets with same
15 or overlapping OTS in the delay buffer. As the packets are
ordered in the delay buffer, output of the "old" packets be-
fore the audio delay change request would be continued nor-
mally, until all data that were in the delay buffer at the
time of change request are output. Then the next packets
20 will indicate for 0.3 sec output times OTS that are already
passed, so the output driver stage will have to discard
these packets.

Another method to handle this case is to write no further
25 data into the output delay buffer beginning with the change
request for the difference time of 0.3 sec, and to use the
new delay time for the OTS calculation for all following
packets. In this case, the output stage of the encoder will
find a more or less continuous OTS sequence.
30
Thus the net effect of delay reduction for the user (con-
sumer listening at the decoder side) will be, that after the
delay change request
- the audio program does continue normally for the present
35 delay time of 0.5 sec,
- then the program continues normally with the new delay of
0.8 sec, but with a delay difference of 0.3 sec of audio
program skipped.

On the operator side, upon change request the following hap-
pens
- all program parts already input to the encoder will be se-
5   quentially, i.e. normally delivered to the user,
- the program material supplied to the encoder immediately
after the change request for a following time period equal
to the delay time difference of 0.3 sec will not be audible
at the user side,
10   - after this time period of 0.3 sec all audio program parts
input to the encoder will be normally audible at the user
side.

Thus, the operator could use a program content switch to
15   change delay time, and simply delay start of the next pro-
gram by the delay time difference in order to guarantee
that nothing of the program is lost for the listener.

All discontinuities of the audio program, that become audi-
20   ble to the listener, might be optionally softened by the
encoder, i.e. proper fade in and out. Delay increase: fade
out before the gap, fade in after the gap. Delay reduction:
fade out before skipped part, fade in after that. In case of
doing the delay changes together with audio program
25   switches, this might not be necessary, because the audio
programs will contain this switch.

The inventive method can be used in an audio encoder as
shown in Fig. 2. The encoder receives four stereo PCM input
30   signals PCMA, PCMB, PCMC and PCMD. E.g. MPEG audio data are
frame based, each frame containing 1152 mono or stereo sam-
ples. The encoder operating system of Fig. 2 may include six
DSPs (not depicted) for the encoding of the four MPEG chan-
nels. These DSPs form a software encoder which includes the
35   technical functions depicted in Fig. 2. A suitable type of
DSP is for example ADSP 21060 or 21061 or 21062 of Analogue
Devices. As an alternative, the technical functions depicted
in Fig. 2 can be realised in hardware.

Synchronisation of the software running on the six DSPs, or
on corresponding hardware, is achieved using FIFO buffers
wherein each buffer is assigned to one or some specific
5 frames. This means that at a certain time instant a current
frame as well as previous frames, the number of which de-
pends from the quantity of available buffers, are present in
the processing stages.

10 Between some of the stages asynchronous buffers ASBUF are
inserted which allow asynchronous write and read operations.
Between other stages synchronous buffers BUF are sufficient.
The PCM input signals PCMA, PCMB, PCMC and PCMD each pass
via an asynchronous buffer to a respective converter CONA,
15 CONB, CONC and COND. In such converter an integer-to-
floating representation conversion of the audio samples to
be encoded may take place. It is also possible that the en-
coder processes integer representation audio samples.
In such converter also one or more kinds of energy levels in
20 a frame may be calculated, e.g. energy of all samples of the
frame or average energy of the samples of a frame. These en-
ergy values may be used in the subsequent psychoacoustic
processing.

25 In addition, in such converter the possibly adapted encoding
parameters can become linked with the frame audio data. In
respective parameter encoders PENCA, PENCB, PENCC and PENCD
the original encoding parameters may be converted as de-
scribed above and then fed to CONA, CONB, CONC and COND, re-
30 spectively.

Via asynchronous buffers the output data of CONA, CONB, CONC
and COND are fed in parallel to subband filters SUBA, SUBB,
SUBC and SUBD and to first left and right channel psy-
35 choacoustic calculators Psycho1A_L, Psycho1A_R, Psycho1B_L,
Psycho1B_R, Psycho1C_L, Psycho1C_R, Psycho1D_L and Psy-
cho1D_R, respectively. The subband filters divide the total
audio spectrum into frequency bands, possibly using FFT, and

may calculate the maximum or scale factor of the coeffi-
cients in a frequency band or subband. Within the frequency
bands a normalisation may be carried out. The subband fil-
ters take into account the above time stamp information and
5   possibly the relevant encoding parameters read from the cor-
responding upstream asynchronous buffer.

The first psychoacoustic calculators perform an FFT having a
length of e.g. 1024 samples and determine the current mask-
10   ing information. Each first psychoacoustic calculator can be
followed by a second psychoacoustic calculator Psycho2A_L,
Psycho2A_R, Psycho2B_L, Psycho2B_R, Psycho2C_L, Psycho2C_R,
Psycho2D_L and Psycho2D_R, respectively, which evaluates the
maximum or scale factor values previously calculated in the
15   subband filters. The first and second psychoacoustic calcu-
lators take into account the above time stamp information
and possibly relevant encoding parameters read from the cor-
responding upstream asynchronous buffers.

20   The output signals of Psycho2A_L, Psycho2A_R, Psycho2B_L,
Psycho2B_R, Psycho2C_L, Psycho2C_R, Psycho2D_L and Psy-
cho2D_R are used in bit allocators and quantisers Bal/Q/E_A,
Bal/Q/E_B, Bal/Q/E_C and Bal/Q/E_D, respectively, for deter-
mining the number of bits allocated and the quantisation the
25   audio data coefficients coming from the associated subband
filter via a buffer. It is also possible to calculate in the
second psychoacoustic calculators in addition what is being
calculated in the first psychoacoustic calculators and
thereby to omit the first psychoacoustic calculators.
30

Finally, the outputs of Bal/Q/E_A, Bal/Q/E_B, Bal/Q/E_C and
Bal/Q/E_D pass through an asynchronous buffers and output
interfaces AES-EBU_A, AES-EBU_B, AES-EBU_C, AES-EBU_D, re-
spectively, which deliver the encoder stereo output signals
35   PCM_Out_A, PCM_Out_B, PCM_Out_C, PCM_Out_D, respectively.
The output interfaces may correspond to IEC 958.

A video encoder includes the following stages: block difference stage, DCT (discrete cosine transform), quantisation and in the feedback loop inverse quantisation, inverse DCT, motion compensated interpolation the output of which is input to the block difference stage, wherein the output of the quantisation is possibly VLC (variable length coding) encoded and buffered before final output and the buffer filling level is used to control the quantisation in such a way that encoding artefacts are masked as far as possible.

In this encoder the following elements are required:
- a system time base supplying a system time, that delivers unique values for all comprised input and output stages;
- hardware and/or software mechanisms that relates system time base and input data to obtain sufficiently precise input time stamps (ITS);
- hardware and/or software mechanisms that relates the system time base with the data output to obtain sufficiently precise output according to the output time stamps (OTS).

These elements are used in the following way:
a) Per input interface of the system, input data are related to the system timer, i.e. an input time stamp ITS along with the incoming data is obtained and assigned to the data frames. For example, the system time of the sampling moment of the first sample of a sampled audio data block or frame is used therefore.
b) In the case of multiple inputs, the input data blocks can be realigned across the channels by the time information given by the input time stamps.
   Example 1: multichannel audio inputs distributed over several two-channel interfaces.
   Example 2: bitstream outputs of multiple stereo encoders shall be multiplexed into an MPEG TS (transport stream) with a well-defined time relation across the channels, i.e. equal delay possibility.
c) From input time stamps ITS and the intended overall delay D output time stamps OTS for the output data are

calculated.

In the easiest processing case where one output data
block per input data block is calculated by the system,
the intended output time of each output block is given
by OTS(n)= ITS(n) + D , n = 0,1,2... denoting the data
block number.

In the case where per input data block several output
data blocks or even a non-integer number of output
blocks is to be generated, the OTS for each of the out-
put blocks can be interpolated according to the corre-
sponding time ratios. An example is the MPEG encoder
1152 samples input, or an MPEG PES packet with one or
more ESPs (elementary stream packets), and the MPEG TS
transport stream wherein the packets have a length of
188 Bytes, i.e. for each sample frame 3 to 7 TS packets
are required for transmission.

d)   Each output interface examines the output data blocks,
      as supplied to the output buffer by the processing
      stages, for their related OTS in relation to the present
      system time as described in detail in the above descrip-
      tion of figure 1. If OTS is indicating a time instant
      that has already passed the output data block can be
      discarded or can be output immediately, depending on the
      application. If the OTS is pointing to a future time in-
      stant the output stage will wait until that time instant
      has come and during the wait time can either output
      nothing or a defined filling pattern.

A couple of related mechanisms are available which can be
used in different combinations as required for the I/O proc-
ess to be implemented.

In an example minimum hardware scenario the system makes use
of a single hardware timer, which is typically already part
of each DSP, in combination with some kind of regular or
controlled output driver software activation. The rest of
the delay control is then executed by the DSP. In principle
two timer functions are required:

o A 'getTime()' function, that allows the software to ask

for the actual system time. Upon reception (DMA or INT based) of the beginning or the end of each input data block the getTime() function can be used to obtain the ITS for that block.

5   ⊚ The output function may need a certain delay before sending an output block for which the corresponding processing has finished. This can either be done in a polling manner, i.e. by cyclic control of the actual system time vs. OTS if some kind of cyclic activation is available, or by a

10   specific timer-based delay function that issues an interrupt after a definable delay 'interruptAfter(DELAY)' or 'threadActivityAfter(DELAY)' function.

The single hardware timer, operating as a backward counter with interrupt at zero, and the input and output DMA block-

15   complete interrupts which are build-in functions of the DSPs, are used.

The single hardware timer can provide the functions 'interruptAfter(DELAY)' and 'getTime()', wherein for the latter function the subsequent delay times loaded to the timer are

20   summed up to obtain a continuous system time and wherein several 'interruptAfter(DELAY)' functions can run in parallel.

In the case of a multi-DSP system with each DSP implementing

25   it's own timer, but distributing inputs and outputs with delay requirements between them across different DSPs, there is a problem of timer synchronicity. This problem can be solved by a special cyclic interrupt signal (e.g. of 10ms length) that is applied to all DSPs in the system and that

30   is used to resynchronise the system times. The counter output word may have the format iiii.ffff, wherein iiii is interpreted as integer part and ffff is interpreted as fractional part. Every 10ms iiii is incremented by '1'. This event is transmitted to the DSPs and is counted therein. The

35   maximum manageable value for DELAY depends on the word length of iiii. Thereby the interrupt indicates the moment of resynchronisation and during the interrupt period a master value iiii for the time is transmitted from one master

DSP to all other DSPs. At this time instant ffff is set to zero.

While the delay control described in above example scenario
5 requires minimum hardware only and thus leaves most of the job to (cheaper and more flexible) software, the disadvantage is that the delay time accuracy is limited by e.g. interrupt latency times, maximum interrupt disable times and, in case of a multi-DSP system, bus arbitration delays.
10 If the achieved accuracy is not sufficient, hardware enhancements can be used that make the process of obtaining ITS relative to system time and output at OTS relative to system time more accurate. A combined solution with software determining the rough point in time and special hardware es-
15 tablishing the exact point in time will achieve a compromise between required DSP reaction time (tends to be slow compared to hardware) and hardware complexity (tends to be more complex the longer the time periods are).

20 A theoretical overflow beginning at the buffers related to the last processing stages and proceeding to the buffers related to the front processing stages is prevented by handshake between the DSPs.

25 The inventive method for changing the delay can also be applied for single DSP systems and for any other kind of real time processing.

In simplified applications, e.g. in an AC-3 decoder, comput-
30 ing a single data frame after reception with the real time constraint that the processing time for each block must be shorter than the frame period is an often used approach. This approach can be extended to a solution with more distributed processing including splitting into several subse-
35 quent processing stages, eventually distributed over several DSPs. In this case each processing stage can be forced into a well defined 'time slot', where for each processing stage the processing time must be shorter than the time slot

length. In contrast to the solution described first, instead
of a single time constraint there would be one time con-
straint per time slot/processing stage.

5 A requirement may be that the encoder is able to operate
   with different encoding parameters because MPEG allows e.g.
   various sample frequencies and overall data rates.

   The invention has the advantage that the shortest disruption
10 period is ensured, when an operator changes the delay.

   The invention can be used to ensure a fixed but variable en-
   coding delay for any variable bit rate encoder or encoder
   which need a variable processing time.
15
   The invention can especially be used for MPEG 1, 2 and 4
   Audio encoding and decoding for MPEG layers 1, 2 or 3, Digi-
   tal Video Broadcast DVB, for AC-3, MD and AAC processing,
   for DVD processing and Internet applications concerning
20 audio data encoding and decoding.